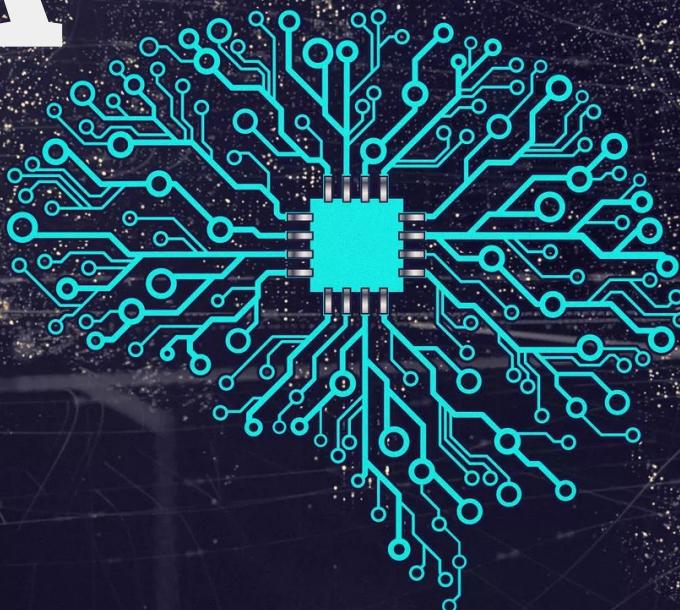
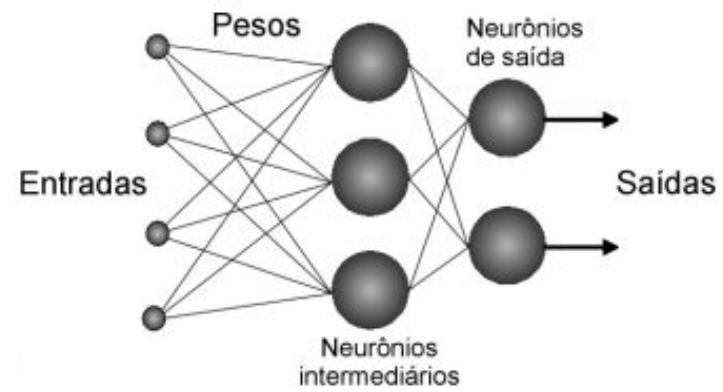
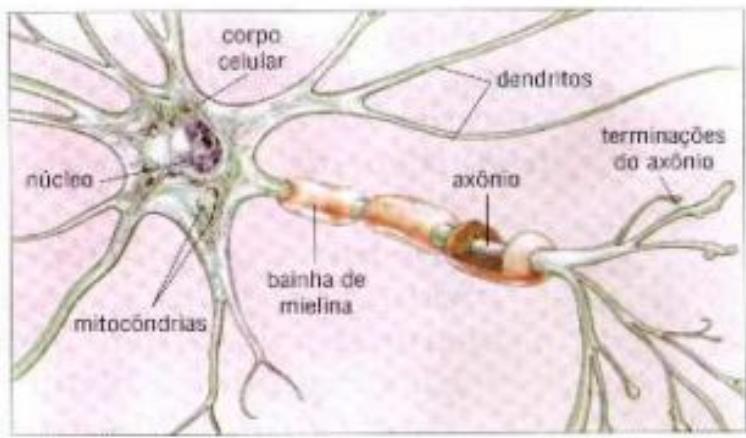


RNA

Redes Neurais Artificiais

*Aula 02





14+

THE ROAD SO FAR

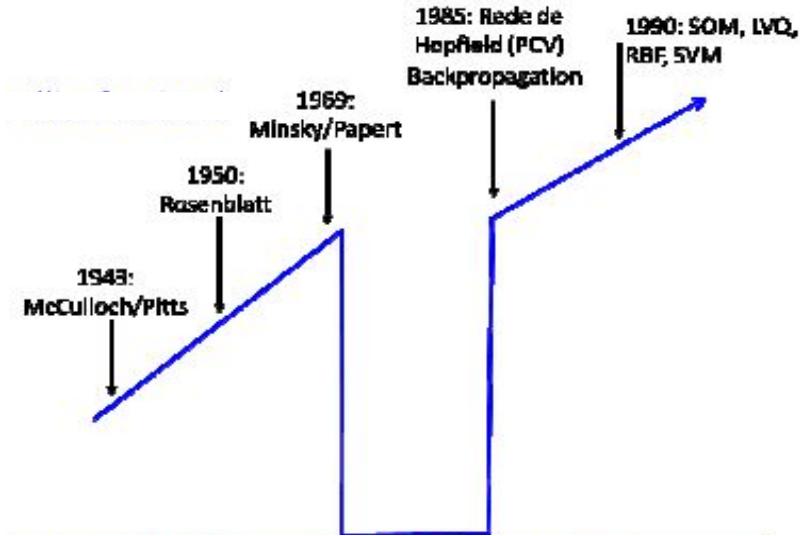


Sinônimos

- sistemas neurais artificiais;
- conexionismo;
- sistemas adaptativos;
- neurocomputadores;
- sistemas paralelos distribuídos

histórico

- paradigmas básicos
 - simbólico
 - conexionista
- **perceptron:** 1 camada de pesos ajustáveis-1957;
- descrédito a partir do final da década de 60;
- em 1974, Werbos lança bases para o **Backpropagation**;
- impulso a partir da década de 80:
 - 1985, com a rede de **Hopfield**, aplicada ao PCV;
 - 1896, com o Backpropagation
- pouco formalismo matemático em muitas áreas.



histórico

Paradigma	Exemplos	Princípios constitutivos	Características naturais				
			Determinismo	Generalização	Reconhecimento	Criação	Autonomia
Paradigma Simbólico	Calculadora, Workflows, Navegação por GPS, Xadrez, Chatbots	Regras lógicas e simbólicas	Alto grau de acurácia e baixíssima incerteza	Capacidade de inferir conceitos genéricos a partir de análise de similaridades e diferenças	Capacidade de classificar algo corretamente mesmo em condições imprecisas	Capacidade de rearranjar elementos na busca de um objetivo, gerando assim algo novo	Capacidade de aprender a partir de dados de campo com o mínimo de intervenção humana, seja na preparação dos dados, seja na configuração de parâmetros
Conexionista	Carros autônomos, reconhecimento de: texto manuscrito, voz, faces, imagens	Funcionamentos dos neurônios	Médio / Baixo	Média / Alto	Alto / Altíssimo	Média	Média

projeto RNA

- definição do problema;
- escolha das informações:
 - obtenção dos dados; criação de arquivos da rede;
- treinamento da rede;
- testes da rede;
- adaptação para uso no problema definido;

razões p/ RNA

- paralelismo;
- capacidade de adaptação;
- memória distribuída;
- capacidade de generalização;
- facilidade de construção;
- desempenho depende da qualidade do pré-tratamento dos dados

aplicações RNA

- regras desconhecidas de resolução de um problema ou difíceis de formalizar;
- problemas com conjuntos de exemplos e suas soluções;
- necessidade de grande rapidez na resolução do problema (respostas em tempo real);
- não existem soluções tecnológicas atuais;
- reconhecimento de formas;
- tratamento de sinal;
- visão, fala;
- previsão e modelagem;

aplicações RNA

- auxílio à decisão;
- problemas de otimização [Siqueira, Scheer, Steiner, 2005, 2006, 2007];
- robótica
- diagnóstico médico [Bennett e Mangasarian, 1990], [Steiner, Carnieri, 1994];
- predição de falência bancária [Tam et al, 1992];
- aplicação à fabricação da pasta e papel industrial [Fadum, 1993]
- controle do processo de produção do papel industrial [Rudd, 1991], [Steiner et al, 1994];
- mundo financeiro [Business Week, 1993], [Cipra, 1992]
- controle de processos químicos [Nascimento et al, 1993]
- obtenção de um modelo organizacional [Almeida, 1995];
- detecção de fraudes com cartões [Financial Times, 1993]
- problemas de administração de empresas [Almeida, 1995]

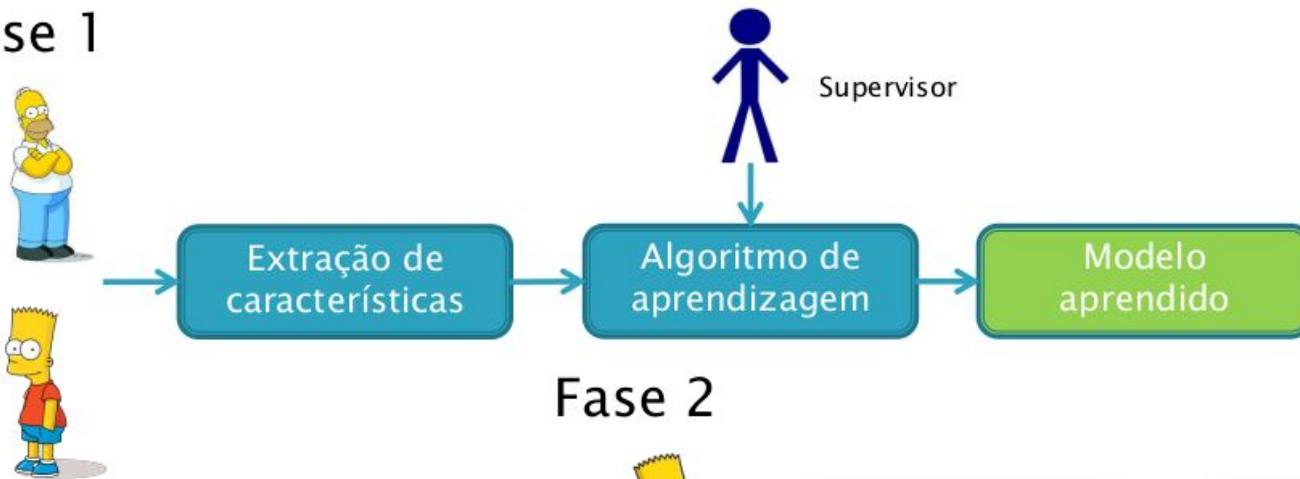
tipos - machine learning

Aprendizagem de Máquina		
Supervisionada	Não supervisionada	Reforço
Classificação	Associação	
Regressão	Agrupamento	
	Detecção de desvios	
	Padrões sequenciais	
	Sumarização	

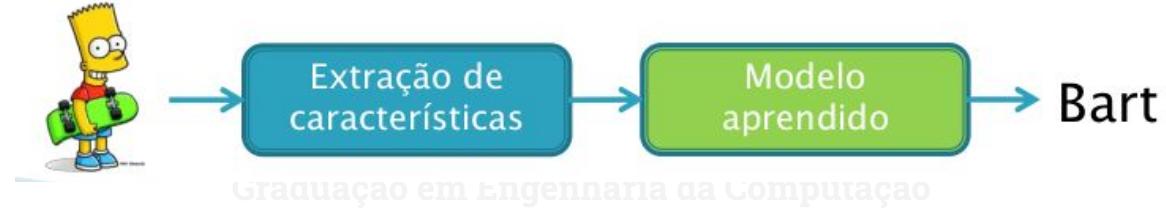
tipos - machine learning

Aprendizagem supervisionada

Fase 1



Fase 2



tipos - machine learning

Classificação (risco de crédito)

História do crédito	Dívida	Garantias	Renda anual	Risco
Ruim	Alta	Nenhuma	< 15.000	Alto
Desconhecida	Alta	Nenhuma	>= 15.000 a <= 35.000	Alto
Desconhecida	Baixa	Nenhuma	>= 15.000 a <= 35.000	Moderado
Desconhecida	Baixa	Nenhuma	< 15.000	Alto
Desconhecida	Baixa	Nenhuma	> 35.000	Baixo
Desconhecida	Baixa	Adequada	> 35.000	Baixo
Ruim	Baixa	Nenhuma	< 15.000	Alto
Ruim	Baixa	Adequada	> 35.000	Moderado
Boa	Baixa	Nenhuma	> 35.000	Baixo
Boa	Alta	Adequada	> 35.000	Baixo
Boa	Alta	Nenhuma	< 15.000	Alto
Boa	Alta	Nenhuma	>= 15.000 a <= 35.000	Moderado
Boa	Alta	Nenhuma	> 35.000	Baixo
Ruim	Alta	Nenhuma	>= 15.000 a <= 35.000	Alto

Treinamento

História do crédito	Dívida	Garantias	Renda anual
Ruim	Alta	Adequada	< 15.000
Desconhecida	Alta	Adequada	< 15.000
Desconhecida	Baixa	Nenhuma	> 35.000
Boa	Alta	Adequada	>= 15.000 a <= 35.000

Teste

tipos - machine learning

Classificação (parafusos)

Comprimento	Diâmetro	Classe
4 cm	3 mm	A
1 cm	1 mm	A
2 cm	2 mm	B
2 cm	3 mm	B
1 cm	2 mm	A
5 cm	3 mm	A
3 cm	3 mm	B

Treinamento

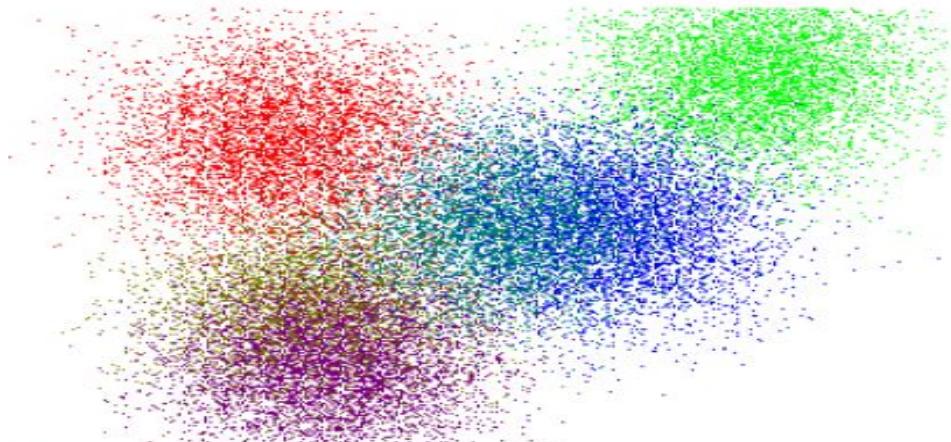
Comprimento	Diâmetro
4 cm	8 mm
7 cm	1 mm
2 cm	2 mm
3 cm	4 mm

Teste

tipos - machine learning

Aprendizagem não-supervisionada

- ▶ Analisar automaticamente os dados (associação, agrupamento)
- ▶ Necessita análise para determinar o significado dos padrões encontrados



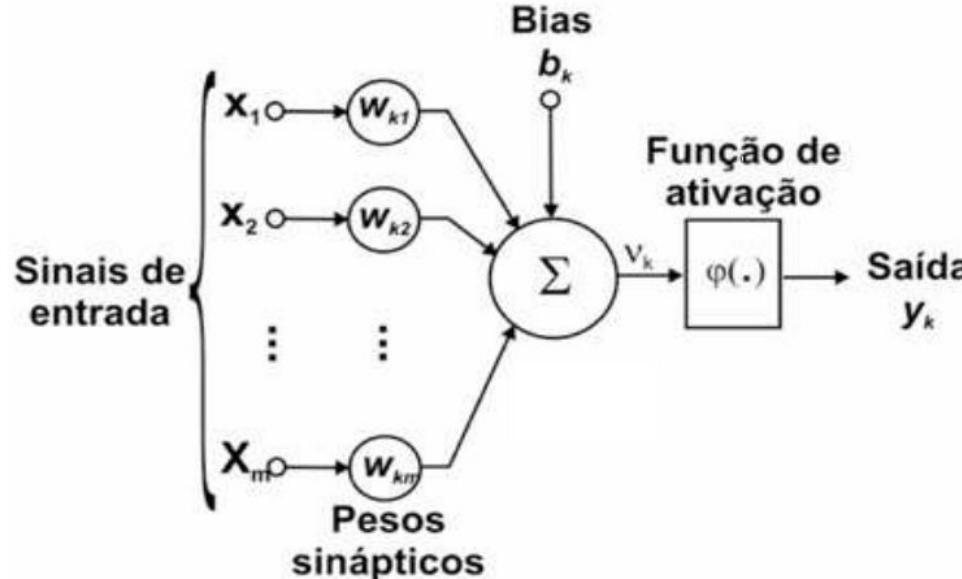
tipos - machine learning

Aprendizagem por reforço

- ▶ Aprender com as interações com o ambiente (causa e efeito)
- ▶ Aprender com sua própria experiência
- ▶ Robô coletando lixo aprendendo a andar em um ambiente
- ▶ Controle automatizado de elevadores

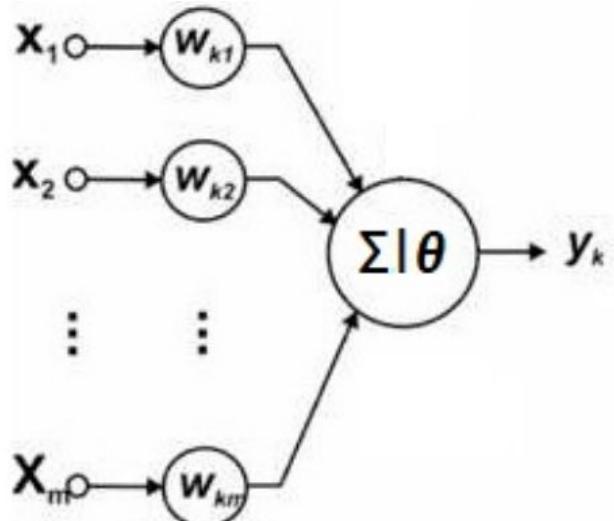
modelo MCP

- Scientific Paper → McCulloch, Warren S. and Pitts, Walter H. **A Logical Calculus of the Ideas Immanent in Nervous Activity.** Bulletin of Mathematical Biophysics, Vol. 5, p. 115 - 133, 1943.
-



modelo MCP

- Neste modelo os neurônios são unidades de processamento simples com uma função de ativação degrau (Limiari).

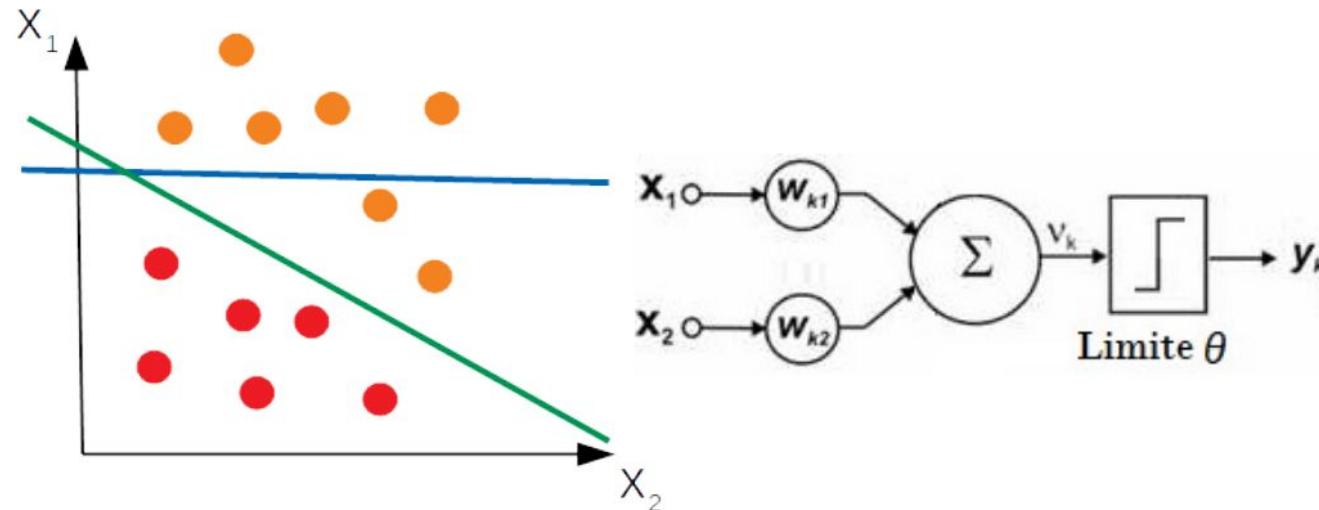


$$Y_k = 1 \text{ se } \left(\sum_{i=1}^n w_{ki} x_i \right) \geq \theta$$

$$Y_k = 0 \text{ caso contrário}$$

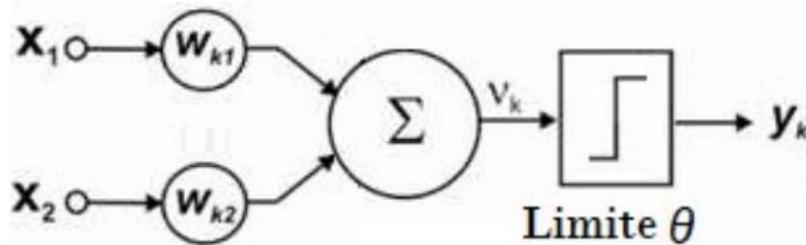
partições espaço de busca

Neurônios com diferentes pesos em suas entradas e diferentes valores de limiar produzem diferentes partições no espaço de entradas.



partições espaço de busca

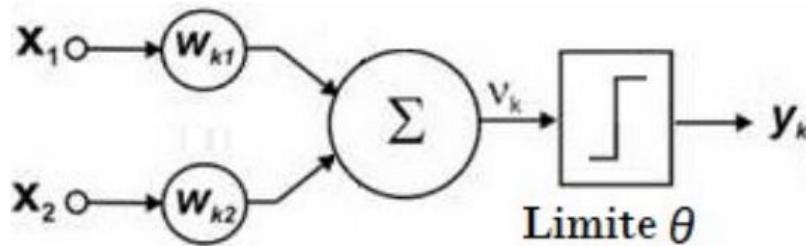
Vamos analisar o caso particular de 2 entradas e 1 saída.



$$Y_k = 1 \text{ se } \left(\sum_{i=1}^n w_{ki} x_i \right) \geq \theta \quad \rightarrow \quad Y_k = 1 \text{ se } W_{k1} X_1 + W_{k2} X_2 \geq \theta$$
$$Y_k = 0 \text{ caso contrário} \quad \rightarrow \quad Y_k = 0 \text{ caso contrário}$$

partições espaço de busca

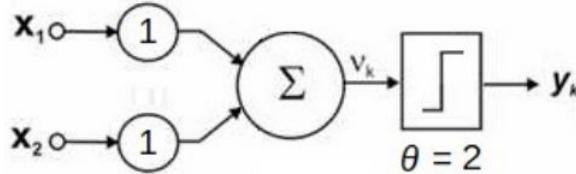
Vamos analisar o caso particular de 2 entradas e 1 saída.



$$Y_k = 1 \text{ se } \left(\sum_{i=1}^n w_{ki} x_i \right) \geq \theta \quad \rightarrow \quad Y_k = 1 \text{ se } W_{k1} X_1 + W_{k2} X_2 \geq \theta$$
$$Y_k = 0 \text{ caso contrário} \quad \rightarrow \quad Y_k = 0 \text{ caso contrário}$$

partições espaço de busca

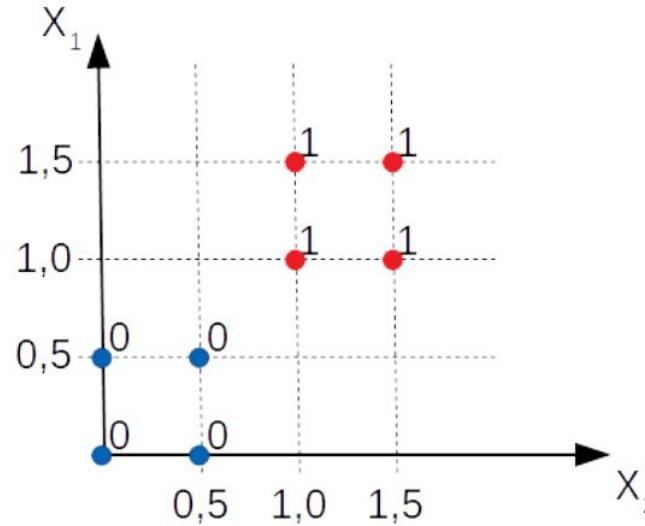
Considerando $w_1 = 1$, $w_2 = 1$ e $\Theta = 2$ temos:



$$Y_k = 1 \text{ se } 1 \cdot X_1 + 1 \cdot X_2 \geq 2$$

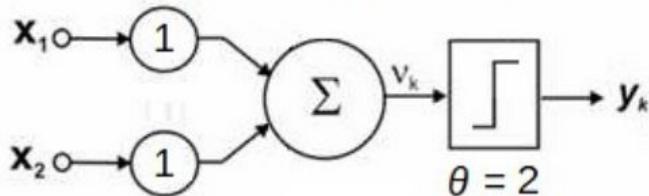
$$Y_k = 0 \text{ caso contrário}$$

X_1	X_2	Y_k
0	0	0
0	0,5	0
0,5	0	0
0,5	0,5	0
1	1	1
1	1,5	1
1,5	1	1
1,5	1,5	1



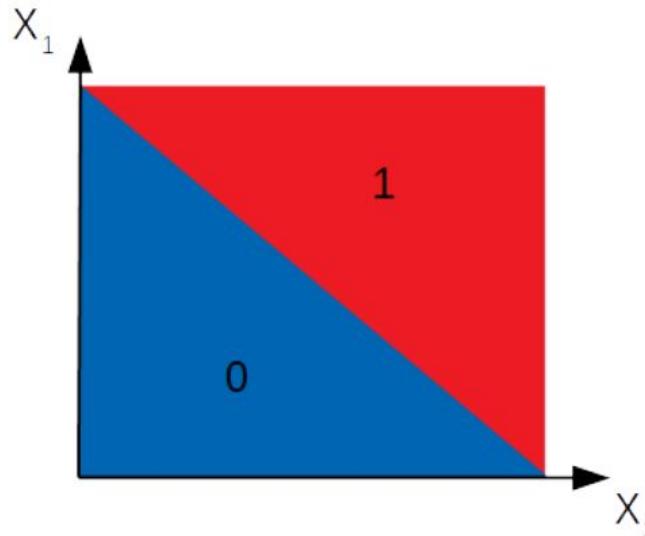
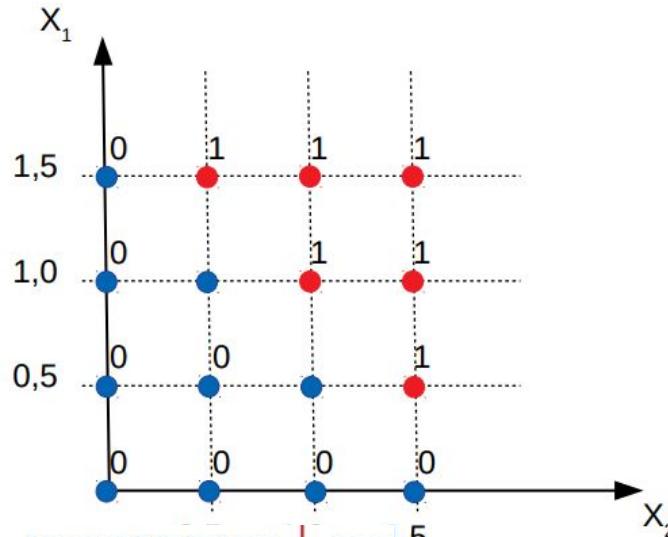
partições espaço de busca

Se aumentarmos o número de entradas temos:



$$Y_k = 1 \text{ se } 1 \cdot X_1 + 1 \cdot X_2 \geq 2$$

$$Y_k = 0 \text{ caso contrário}$$



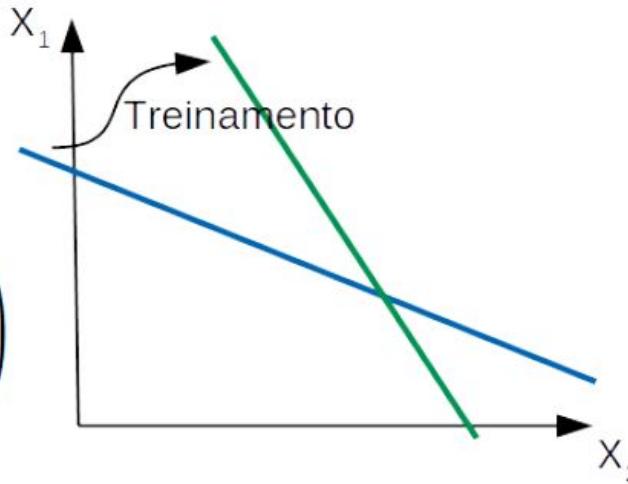
partições espaço de busca

A saída (y) é uma função da combinação linear das entradas (x).

$$W_1 \cdot X_1 + W_2 \cdot X_2 = \theta$$

ou

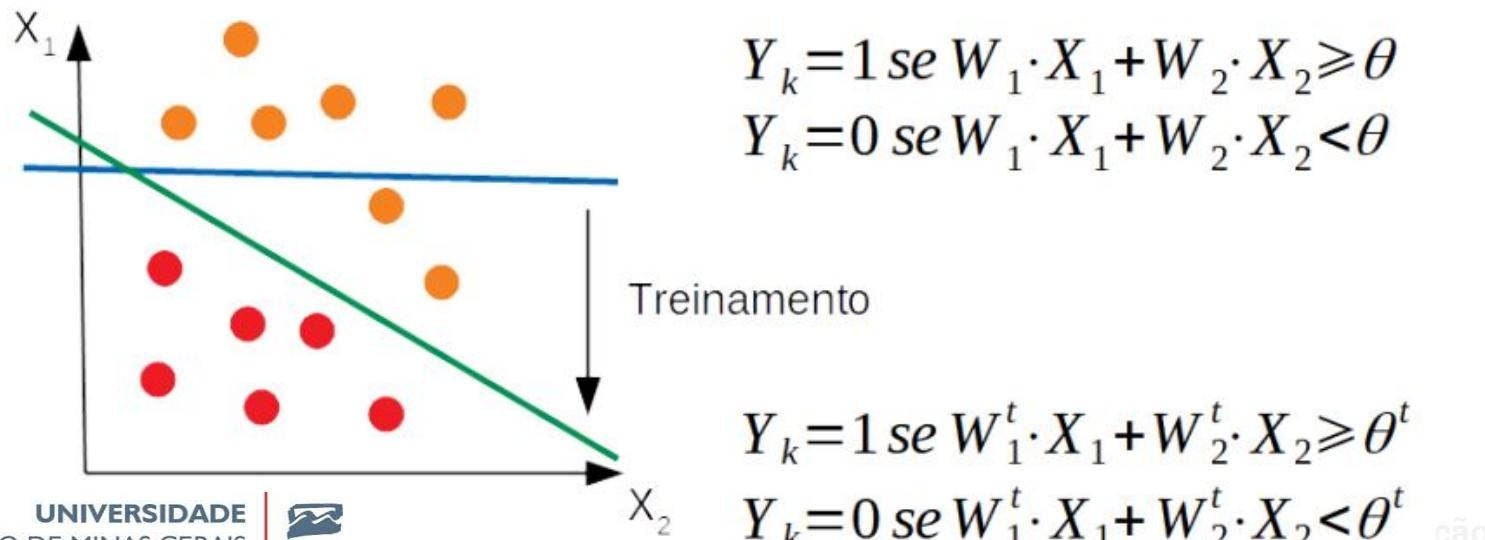
$$X_2 = -\left(\frac{W_1}{W_2}\right) \cdot X_1 + \left(\frac{\theta}{W_2}\right)$$



O treinamento de um neurônio e consequentemente de uma Rede Neural Artificial acontece através da alteração dos parâmetros w_1 , w_2 e Θ .

partições espaço de busca

Levando em consideração que a alteração dos parâmetros w e Θ (treinamento) modifica a posição da reta e portanto da partição no espaço de entrada é possível observar que a estrutura de uma Rede Neural Artificial é apropriada para resolver problemas de classificação.



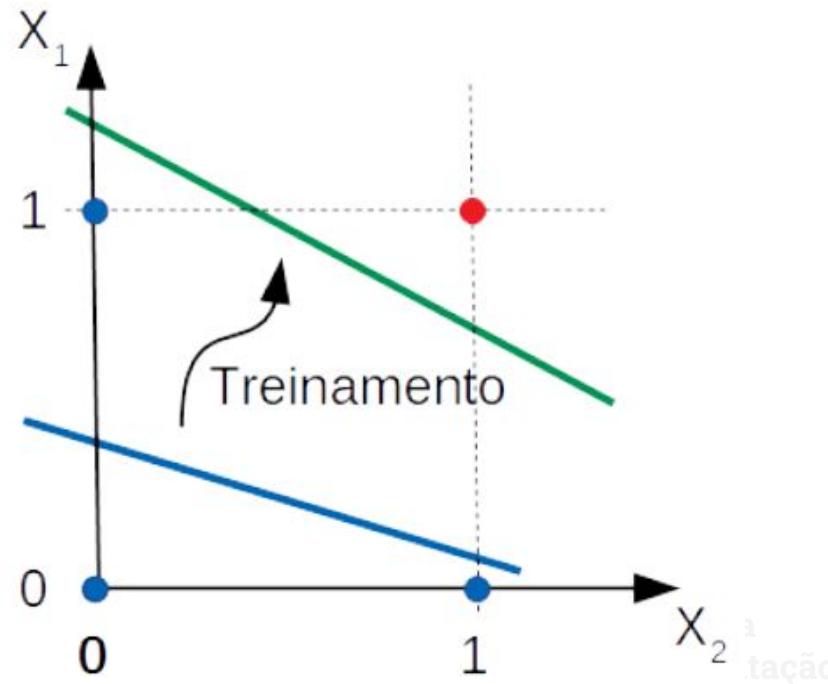
partições espaço de busca

Devido a simplicidade apresentada para os neurônios este modelo ser apenas para a solução de problemas simples ou linearmente separáveis.

Exemplo:

AND Lógico

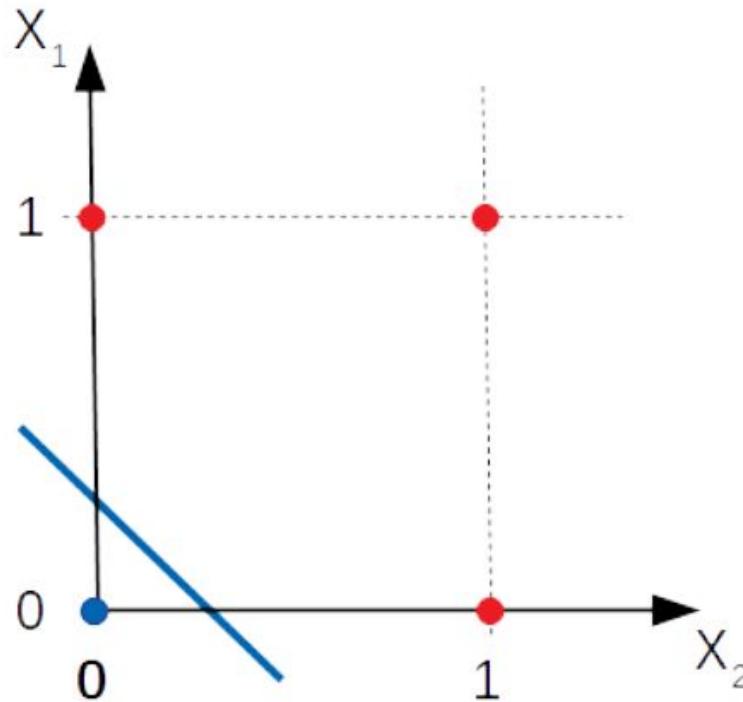
X_1	X_2	Y_k
1	1	1
0	1	0
1	0	0
0	0	0



partições espaço de busca

OR Lógico: Também é linearmente separável.

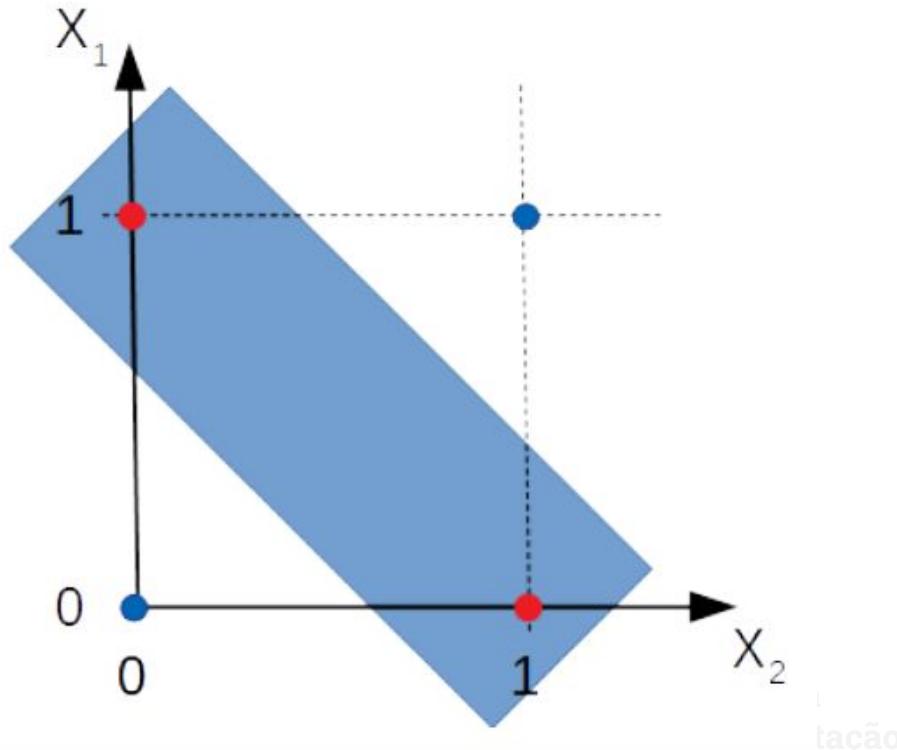
x_1	x_2	y_k
1	1	1
0	1	1
1	0	1
0	0	0



partições espaço de busca

XOR Lógico: Não é linearmente separável.

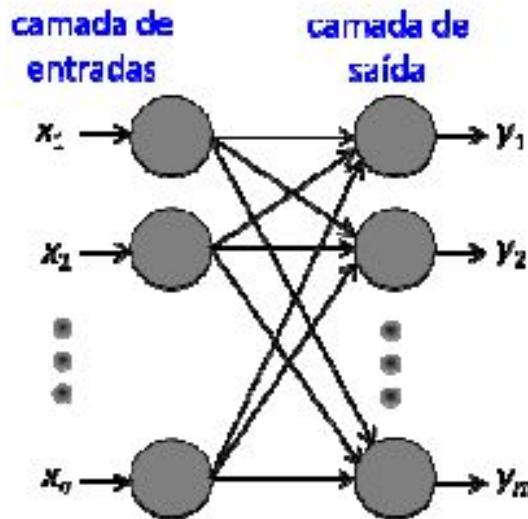
X_1	X_2	Y_k
1	1	0
0	1	1
1	0	1
0	0	0



perceptron

- É um tipo de rede neural usada para conjuntos de treinamento linearmente separáveis [Rosenblatt, 1950].
- Tem a inclusão de bias (tendência)

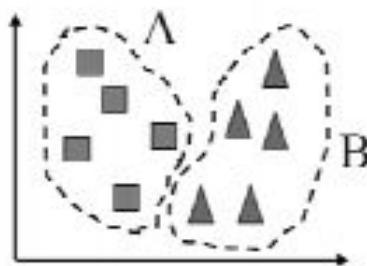
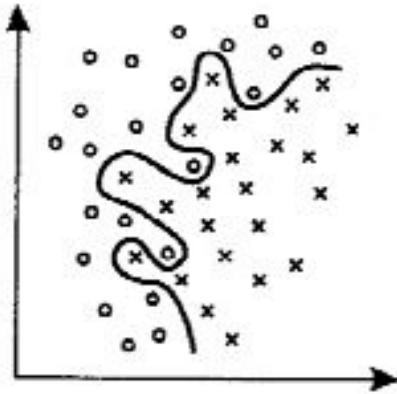
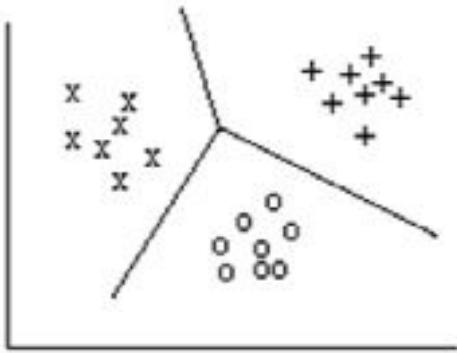
$$z = \sum_{i=1}^N x_i w_i + b$$



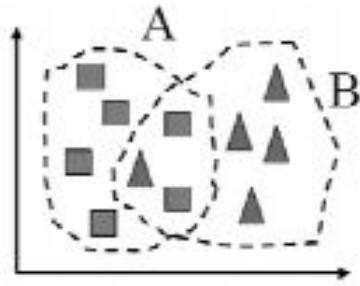
perceptron

- O algoritmo de aprendizagem do **Perceptron** procura um vetor **w** com projeção positiva (produto interno) com todos os **exemplos positivos** e projeção negativa com os **exemplos negativos**.
- A aprendizagem do perceptron sempre tem sucesso em tempo finito para um conjunto de treinamento finito e separável de exemplos de treinamento.

perceptron



← OK
Não OK →



implementação

